REMARKS

The Office action has been carefully considered. The Office action rejected claims 20 and 61 under 35 U.S.C. § 112, second paragraph as being indefinite. Further, the Office action rejected claims 1-3, 18, 20, 28, 36, and 62 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,741,242 B1 to Itoh et al. ("Itoh") in view of U.S. Patent Publication 2003/0120983 to Kim et al. ("Kim"). Further yet, the Office action rejected claims 4-17, 19, 21-27, 30-35, and 37-61 under 35 U.S.C. § 103(a) as being unpatentable over Itoh in view of Kim and in further view of U.S. Patent Publication 2004/0110490 to Steele et al. ("Steele") while also referencing *Scalable Vector Graphics specification version 1.1* revised January 14, 2003 ("SVG"). Finally, the Office action rejected claim 29 under 35 U.S.C. § 103(a) as being unpatentable over Kim in view of Steele and SVG and in further view X3D specification ("X3D"). Regarding the §112 rejections, applicants have amended the claims to obviate these rejections. As will be discussed below, applicants respectfully disagree with regard to the §103 rejections.

By present amendment, claims 20 and 61 have been amended for clarification and not in view of the prior art. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability. Despite the Office action's most recent comments characterizing the conversation held by telephone on March 15, 2005, applicants still maintain that the claims as filed were patentable over the prior art of record. Applicants further assert that amendments herein remain for

purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a new element object model and a vector graphics markup language for accessing element object models in a manner that allows program code developers to consistently interface with a scene graph data structure to produce graphics. The vector graphics markup language may comprise an interchange format for expressing vector graphics via the element object model. When interpreted, the markup may be parsed into data including elements in an element tree that is translated into the objects of a scene graph data structure. At the element tree level, a property system and layout system that may provide rich programmability features, including inheritance characteristics and eventing, making it straightforward for scene designers to design possibly complex scenes. In general, the vector graphics elements may correspond to shape elements and other elements including image and video elements that correlate with scene graph objects of the scene graph object model. The properties and other resources of the vector graphics elements may also correlate with similar properties and resources the scene graph object model.

The vector graphics system may also program to an element level, in which each of the drawing shapes is represented as an element at the same level as the rest of the programmable elements in a page/screen, allowing interaction with the

layout system, events and properties. The vector graphics system also may provide a mechanism for programming to a resource level, by which scene designers may essentially shortcut the layout system and program directly to the visual application programming interface layer that interfaces with the scene graph data structure. This alternative often provides a more efficient and lightweight way to output the appropriate data.

In one implementation, the markup code may be interpreted by a parser / translator which generally may add element-level elements to an element tree / property system and attaches data to those elements. The layout system then may take the element tree with the attached presenters and may translate the data to objects (via a builder) and calls to a visual application programming interface layer that may interface with the scene graph and creates the scene graph objects.

Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

Rejections under §103

Turning to the §103 rejections, claim 1 recites in a computing environment, a computer-implemented method comprising receiving a function call via an application programming interface, the function call comprising markup language data, interpreting the markup language data to cause data in a scene graph to be modified, and causing a change in a display in response to the modification of data in the scene graph.

The Office action rejected claim 1 as being unpatentable over Itoh (with reference to SVG) in view of Kim. More specifically, the Office action contends that

the prior art of record teaches receiving a function call via a programming interface, the function call comprising markup language data. Figs. 1-3, and paragraphs 0004 to 0008 of Kim are referenced as well as column 1, line 20 to column 2, line 25 and column 19, lines 1-20 of Itoh. Further, the Office action contends that the prior art of record teaches interpreting the markup language data to cause data in a scene graph to be modified. Paragraphs 0007, 000, 0020, 0026 and Fig. 4 of Kim as well as columns 1 and 2 of Itoh are referenced. Finally, the Office action contends that the prior art of record teaches causing a change in a display in response to the modification of data in the scene graph. Fig. 4 of Kim is referenced.

It had been acknowledged in the previous Office action that Kim does not teach the modification of data in the scene graph. Furthermore, the current Office action acknowledges that Itoh does not teach interpreting markup language by parsing the markup. The Office action, however, contends that the Itoh reference together with the Kim reference teach all of the recitations of claim 1 in some way shape or form and that because Kim and Itoh teach analogous art, a person skilled in the art at the time the invention was made would have found obvious the recitations of claim 1. Applicants respectfully disagree.

It is well settled law that to establish *prima facie* obviousness of a claimed invention, all of the claim recitations must be taught or suggested by the prior art; (*In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)), and "all words in a claim must be considered in judging the patentability of that claim against the prior art;" (*In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)).

Further, if prior art, in any material respect teaches away from the claimed invention, the art cannot be used to support an obviousness rejection. *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed Cir. 1997). This remains a basic tenet of patent law jurisprudence despite the characterization by the Office action that teaching away is subject to interpretation in view of the circumstances on a case-by-case basis. Moreover, if a modification would render a reference unsatisfactory for its intended purpose, the suggested modification / combination is impermissible. See MPEP § 2143.01. As will be shown below, the Office action has failed to establish a *prima facie* case for obviousness.

As was presented in the previous Office action response, Kim is directed to a system for implementing a 3D virtual reality by using extensible 3D (X3D) data provided form a service server. In particular, the cited and applied sections of Kim teach a server that includes an applet for communicating with the server to receive X3D data, a communication module connected to the applet for providing a communication between the server and a client, and a browsing means for parsing the X3D data to construct a scene graph, rendering the scene graph, and displaying the scene graph. However, the cited and applied sections of Kim teach a broad overview of virtual reality system that may include such "functions" as collision detection Walk, Slide, and Look. These functions as used in Kim are not function calls as used in the present invention and certainly cannot be construed to be a function call via an application programming interface.

Itoh, likewise, is directed to a system for integrating and displaying multimedia documents. In particular, the system of Itoh utilizes a framework

function section for operating with a parser wherein the system can interpret various drawing tags in a document described in a markup language. By parsing the markup, the system of Itoh then generates a framework for synchronizing and operating various drawing primitives and generates a two-dimensional drawing command and a three-dimensional drawing command based on the framework. In this manner, two-dimensional objects and three-dimensional objects may be rendered in a web browser more easily. Itoh, however, is silent with respect to any method or system in which a function call in a markup language is initiated via an application programming interface (API).

Quite different from the prior art of record, claim 1 recites receiving a function call via an API. It is well established in the computer arts that an API may comprise a set of routines used by an application program to direct the performance of procedures, for example, by a computer's operating system. In this embodiment, the API is operable to receive a function call that comprises markup language data. Such a function call that is in a markup language provides an interchange format for expressing vector graphics via the element object model.

The Office action contends that Itoh teaches receiving a function call via an API, the function call comprising markup language data. This is an inaccurate interpretation of the teachings of Itoh. Specifically, Itoh concedes that a two-dimensional drawing command of Java AWT prepared as a Java applet is the only drawing command capable of a graphics display on its Web browser without unnecessary installation on the client personal computer. See column 19, lines 7-11 of Itoh. Therefore, any three-dimensional rendering function section is

structured to internally process a mechanism of a three-dimensional graphics pipeline and call an API of Java AWT for a final drawing command. See column 19, lines 11-14 of Itoh. Thus, Itoh teaches that when three-dimensional graphics are required, a separate applet must be called via an API for the applet. Itoh goes on to teach that the three-dimensional rendering function receives a three-dimensional drawing command, which indicates an attribute, a behavior, or a setting, from a rendering function section. The three-dimensional drawing command then generates graphics display data for expressing a drawing in three dimensions with a command set for a final two-dimensional drawing of Java AWT. The graphics display data is then passed to the API of Java AWT provided by a Web browser. Thus, to get from function call to the API, Itoh teaches taking a drawing command (which is not in a mark-up language) to generate graphics display data (which is still not in a mark-up language) which may then be passed to an API of an applet for then interpreting the graphics display data to be interpreted by a web browser. See column 19, line 51 to column 20, line 3 of Itoh.

Handling a function call in a first format (graphics command format) and an interim second format (graphics display format) and then via an API to yield data in a markup language is far different than a function call in a markup language received via an API as recited in claim 1. For the sake of argument, even if the two formats described above may be considered a markup language of sorts, the fact remains that Itoh teaches a transition with two interim steps that the recitations of the present invention do not have. Simply put, Itoh does not teach or even suggest the method of the present invention as recited in the system of claim 1.

In fact, Itoh teaches a system having the very problem that is solved by the present invention. The reason that the function call of Itoh is manipulated so much is that the Java applet of Itoh is a conventional Java applet that cannot handle a function call received in a markup language. The function call must necessarily be translated to a format suitable with the application programming interface of the Java applet. Translating data that already exists in a markup language into a format other than a markup language format for the specific purpose of engaging an API fits squarely within the very definition of teaching away as generally espoused in *In re Geisler*. Clearly, then, Itoh may not be used in the manner suggested in the Office action.

As such, applicants maintain that nowhere in any prior art of record can there be found a teaching or suggestion of a function call in a markup language received via an API; in fact at least Itoh teaches directly away from the claimed subject matter. Applicants submit that the Office action has failed to establish a *prima facie* case of obviousness as all of the language in the recitations has not been either disclosed or suggested by the prior art of record.

For at least the foregoing reasons, applicants submit that claim 1 is allowable over the prior art of record because Itoh and Kim, whether considered alone or in any permissible combination with each other or any other prior art of record, does not teach or suggest the recitation of claim 1.

Applicants respectfully submit that dependent claims 2-35, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 1 and consequently includes the recitations of independent claim 1. As

discussed above, Kim and Itoh, whether considered alone or in any permissible combination with each other or other prior art, such as Steele, fail to teach or suggest the recitations of claim 1 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 1 noted above, each of these dependent claims includes additional patentable elements.

For example, claim 2 recites wherein causing data in the scene graph to be modified comprises causing initialization of a new instance of a visual class. The Office action contends that both Itoh and Kim implicitly teach this recitation. However, nowhere in Itoh or Kim is there any mention of instances or classes. It stands to reason that if Itoh and Kim used such concepts, then each would say so explicitly. The Office action is broadly and improperly concluding that Itoh and Kim are implicitly teaching far beyond that which Itoh and Kim actually encompass. Applicants submit that claim 2 is allowable over the prior art for at least this additional reason.

Turing to the next independent claim, claim 36 recites in a computing environment, a computer system comprising a markup parser, an application programming interface coupling the markup parser to a source of markup, a container for visual information of an object model, the markup parser converting markup received at the application programming interface to method calls of objects in the object mode to modify data in the container for visual information, and an video interface operable to interpret the visual information for display on a display.

The Office action rejected claim 36 as being unpatentable over Kim in view of Itoh. More specifically, the Office action incorporated the rejection of claim 1 in its entirety into the rejection of claim 36 and, thus, cites similar references to the prior art to support the contended rejection. Applicants respectfully disagree.

In one recitation, claim 36 recites a markup parser for converting markup received at the API into method calls of objects in the object mode to modify data in the container for visual information. Thus, the API is operable to receive a function call that comprises markup language data. Such a function call that is in a markup language provides an interchange format for expressing vector graphics via the element object model.

Much like the rejection of claim 1, the Office action contends that Itoh teaches a markup parser for converting markup received at the API into method calls of objects in the object mode. Again, this is an inaccurate interpretation of the teachings of Itoh. Itoh teaches handling a function call in a first format (graphics command format) and an interim second format (graphics display format) and only then via an API to yield data in a markup language. This is far different than a function call in a markup language received via an API as recited in claim 36. For the sake of argument, even if the two formats described above may be considered a markup language of sorts, the fact remains that Itoh teaches a transition with two interim steps that the recitations of the present invention do not have. Simply put, Itoh does not teach or even suggest the method of the present invention as recited in the system of claim 36.

In fact, Itoh teaches a system having the very problem that is solved by the present invention. The reason that the function call of Itoh is manipulated so much is that the Java applet of Itoh is a conventional Java applet that cannot handle a function call received in a markup language. The function call must necessarily be translated to a format suitable with the API of the Java applet. Translating data that already exists in a markup language into a format other than a markup language format for the specific purpose of engaging an API fits squarely within the very definition of teaching away as generally espoused in *In re Geisler*. Clearly, then, Itoh may not be used in the manner suggested in the Office action.

As such applicants maintain that nowhere in any prior art of record can there be found a teaching or suggestion of a markup parser for converting markup received at the API into method calls of objects in the object mode. Applicants submit that the Office action has failed to establish a *prima facie* case of obviousness as all of the language in the recitations has not been either disclosed or suggested by the prior art of record.

For at least the foregoing reasons, applicants submit that claim 36 is allowable over the prior art of record because Itoh and Kim, whether considered alone or in any permissible combination with each other or any other prior art of record, does not teach or suggest the recitation of claim 36.

Applicants respectfully submit that dependent claims 37-61, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 36 and consequently includes the recitations of independent claim 36. As discussed above, Itoh and Kim, whether considered alone or in any permissible

combination with each other or any other prior art of record (such as Steele or SVG), fails to teach or suggest the recitations of claim 36 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 36 noted above, each of these dependent claims includes additional patentable elements.

Turning to the last independent claim, amended claim 62 recites In a computing environment, a computer system comprising application programming interface means for receiving function calls comprising markup, parsing means for converting the markup to data corresponding to an object model associated with rendering graphics data, and rendering means for outputting the graphics data to a display.

The Office action rejected claim 62 as being unpatentable over Kim in view of Itoh. More specifically, the Office action once again incorporated the rejection of claim 1 in its entirety into the rejection of claim 36 and, thus, cites similar references to the prior art to support the contended rejection. Applicants respectfully disagree.

Claim 62 recites an application programming interface means for receiving function calls comprising markup. Thus, the application programming interface is operable to receive a function call that comprises markup language data. Such a function call that is in a markup language provides an interchange format for expressing vector graphics via the element object model.

Much like the rejection of claims 1 and 36, the Office action contends that Itoh teaches an application programming interface means for receiving function

calls comprising markup. Again, this is an inaccurate interpretation of the teachings of Itoh. Itoh teaches handling a function call in a first format (graphics command format) and an interim second format (graphics display format) and then via an API to yield data in a markup language. This is far different than a function call in a markup language received via an API as recited in claim 62. Even if the two formats described above may be considered a markup language of sorts, the fact remains that Itoh teaches a transition with two interim steps that the recitations of the present invention do not have. Simply put, Itoh does not teach or even suggest the method of the present invention as recited in the system of claim 62.

In fact, Itoh teaches a system having the very problem that is solved by the present invention. The reason that the function call of Itoh is manipulated so much is that the Java applet of Itoh is a conventional Java applet that cannot handle a function call received in a markup language. The function call must necessarily be translated to a format suitable with the API of the Java applet. Translating data that already exists in a markup language into a format other than a markup language format for the specific purpose of engaging an API fits squarely within the very definition of teaching away as generally espoused in *In re Geisler*. Clearly, then, Itoh may not be used in the manner suggested in the Office action.

As such applicants maintain that nowhere in any prior art of record can there be found a teaching or suggestion of an API means for receiving function calls comprising markup. Applicants submit that the Office action has failed to establish a *prima facie* case of obviousness as all of the language in the recitations has not been either disclosed or suggested by the prior art of record.

For at least the foregoing reasons, applicants submit that claim 62 is allowable over the prior art of record because Itoh and Kim, whether considered alone or in any permissible combination with each other or any other prior art of record, does not teach or suggest the recitation of claim 62.

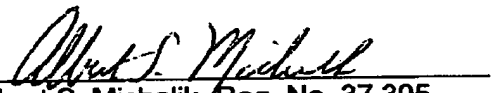
For at least these additional reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and early allowance of this application is earnestly solicited.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-62 are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

In re Application of BEDA et al.
Serial No. 10/693,633

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this Amendment, along with transmittal, petition for extension of time, credit card payment form, and facsimile cover sheet, are being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. 1.6(d) on the date shown below:

Date: September 27, 2005

Albert S. Michalik

3481 Amendment